Toolkit for Bar Code Recognition and Resolving on Camera Phones – Jump Starting the Internet of Things

Robert Adelmann, Marc Langheinrich, Christian Flörkemeier
Institute for Pervasive Computing, ETH Zurich
{adelmann, langhein, floerkem}@inf.ethz.ch

Abstract: Automatic identification technology such as RFID promises to connect physical objects with virtual representations or even computational capabilities. However, even though RFID tags are continuously falling in price, their widespread use on consumer items is still several years away, rendering large-scale experiments with such an "internet of things" difficult. Much more ubiquitous are printed bar codes, yet so far their recognition required either specialized scanner equipment, custom-tailored bar codes or costly commercial licenses – all equally significant deployment hurdles. We have developed a freely available EAN-13 bar code recognition and information system that is both lightweight and fast enough for the use on camera-equipped mobile phones, thus significantly lowering the barrier for large-scale, real-world testing of novel information and interaction applications based on "connected" physical objects. We hope that this "low tech" version of bridging the gap will allow the community to quickly develop and try out more realistic and widespread applications, and thus gain real-world experiences for better jump-starting the future internet of things, today.

1 Today's Role of Barcode Recognition

The idea of *linking real-world products with virtual information* has been around for quite some time. In 1998, Barrett and Maglio already described a system for attaching information to real-world objects [BM98], while 1999 Want et al. expanded upon the idea and linked arbitrary items through the use of RFID tags with both information services and actions [WFGH99]. Since then, a number of research projects have continued to explore this concept of "bridging the gap", i.e., the automatic identification of individually tagged real-world products in order to quickly look up information or initiate a specific action [KBM$^+$02]. With the increasing mobility of powerful computing systems, e.g., mobile phones or handheld PDAs, this bridging can even be done *in situ*, i.e., right when we need it, where we need it.

While RFID potentially offers an unprecedented user experience due to its detailed means for identification (i.e., on a per item basis) and the lack of a line-of-sight requirement for reading, most industry analysts agree that an item-level rollout (e.g., having an RFID tag on every single supermarket product) is still several years away [Jue05]. In contrast, the printed bar codes are practically ubiquitous: Virtually every item sold today carries an internationally standardized bar code on its packaging, enabling not only checkout registers to quickly sum up one's shopping items, but also to identify a product and look up a wealth of related information. Obviously, using bar codes for linking real-world objects to virtual information has a number of drawbacks when compared to an RFID-enabled future with corresponding mobile RFID readers, such as NFC-enabled[1] mobile phones. Due to their sensitivity to soiling, ripping, and lighting conditions, optical bar code recognition can be difficult. Until recently, reading a conventional (i.e., 1D) bar code inevitably required a separate laser scanner or a corresponding mobile phone scanner attachment.

The increasing availability of camera phones, i.e., mobile phones with an integrated digital camera, has begun to simplify this process, however. After 2D bar codes have been successfully recognized by most consumer-grade camera phones for quite some time [Roh04], the continuously increasing quality of both the camera resolution and the employed lenses have finally made it feasible to directly read 1D bar codes with such cameras, without the need for special attachments or handheld lasers. This significantly changes the attractiveness of using barcodes for the above physical-to-digital linkage: Instead of waiting several years for a comprehensive item-level roll out of RFID tags, or forcing people to carry around specific scanner attachments for their mobile phones, the support of 1D bar code recognition on any camera phone immediately allows anybody to interact with almost any commercially available product – all it takes is a small application download.

The main contribution of this paper is a freely available 1D bar code recognition toolkit that is intended to facilitate the creation of novel applications and services. We believe that the adequate performance of our recognition software, when compared with existing commercial implementations, the ease with which external data sources can be integrated, and the availability of our toolkit under an open source license will help to foster the the use of camera phones as mobile bar code scanners.

2 Related Work

Prior work on using printed bar codes for linking real-world objects with virtual information has often used two dimensional bar codes [RA00, PKA05], which do not use bars of varying widths but instead blocky rectangles that lend themselves much better to low resolutions or misalignments. There is a wide variety of code symbologies available, such as Semacodes, Spotcodes, the Japanese QR-System[2], or Rohs' VisualCodes [Roh04]. All of these systems were specifically designed to simplify camera-based recognition. However, while they offer both improved detection rates as well as additional services such as range and alignment detection, none of these codes enjoys widespread use, let alone comes close to the billions of products carrying EAN-13 bar codes today. Also, none of these codes is linked to a wealth of EAN-13-indexed information available in online databases today.

A number of algorithms have already been implemented for the visual decoding of 1D bar codes on desktop computers[3]. Most of these are based on the transformation of the origi-

[1] Near Field Communication (NFC) is a new standard for mobile phones that allows them to both act as an RFID reader and be read by other RFID readers (see `www.nfc-forum.org`). Many handset manufacturer have already begun shipping NFC-enabled models.

[2] See `www.semacodes.org`, `www.op3.com` and `www.qrcode.com`, respectively.

[3] See for example `www.charactell.com/iRead.html` or `www.axtel.com`.

Figure 1: *Multiple scanlines:* In contrast to existing approaches, we make intense use of multiple scanlines in our algorithm to both increase robustness and improve accuracy. While the image shows all of the scanlines oriented in parallel, our system supports any orientation of scanlines.

nal image information into a decoding domain that simplifies bar code identification, like approaches based on the Fourier transformation or the Hough transformation as proposed by Muniz et al. [MJO99]. These approaches are often used in professional image recognition software, as the offer very good recognition rates. However, their requirements in terms of system resources can be too demanding for typical mobile devices. While both Ohbuchi et al. [OHH04] and Chai and Hock [CH05] have presented algorithms intended for mobile devices, these algorithms so far have not been implemented or tested on actual mobile camera phones. As an alternative to costly domain transformation, a much simpler approach is based on so-called scanlines, which try to detect the bar code along a particular line through the image[4]. As such algorithms need much fewer computing resources, they are specifically relevant for the use on mobile camera phones. Their drawbacks, however, lie in their often poor recognition rates when dealing with dirty surfaces, reflections or shadows, or slight misalignments and their need of detecting the bar code in the image first (in order to properly align the scanline). We improved on this by making extensive use of multiple scanlines (see figure 1), which will be explained below in section 3.1.

Note that given the commercial potential of the 1D barcode recognition on mobile phones, it is not surprising that a number of commercial solutions exist. Scanbuy offers an application called ScanBuy Decoder[5], which is capable of recognizing 1D barcodes. Similar applications can be bought from PaperClick[6], Gavitec[7], and MediaStick[8], to name but a few. While informal trials with some freely available beta programs from the above vendors showed a comparable, sometimes even superior performance of our system, we explicitly abstained from conducting formal comparisons, as improving the recognition rate or speed is not our primary goal. Instead, we are trying to create a free, easily usable, and robust barcode recognition system for mobile phones, together with an open resolving framework that facilitates rapid prototyping and deployment. The currently available commercial systems, in contrast, not only restrict source-code access but also typically limit barcode resolving to vendor applications and/or a fixed set of lookup services.

[4] See for example `sourceforge.net/projects/barcr-reader/`.
[5] See `www.scanbuy.com/website/products_decoder.htm`.
[6] See `www.paperclick.com/`.
[7] See `www.mobiledigit.de`.
[8] See `www.mediastick.co.jp`.

3 System Design

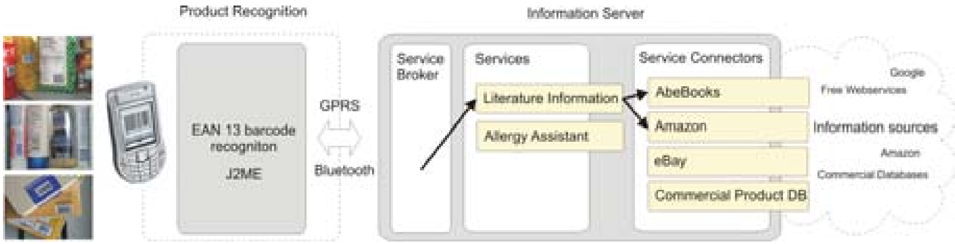

Figure 2: *Architectural Overview:* Our EAN-13 recognition and resolution system consists of a mobile phone application for code recognition, and a server side component for code resolution.

Our EAN-13 bar code recognition and resolution toolkit contains two parts: the barcode recognition component running entirely on J2ME enabled mobile phones that support the MMAPI[9] (Mobile Media APIextension) extension and the Java based information server component, which is located on a separate server, to which the detected product code is transmitted via a GPRS (or for local demonstration a Bluetooth) connection. The provided client provides functionality to recognize an EAN13 code, communicate with the server and display the results. The information server uses a plug-in architecture, allowing us to quickly add various services and online information sources (represented as so called "service connectors"). Although this process could also be located on the phone itself, performing them on an external server provides us with greater extendibility, higher flexibility and better performance.

3.1 Recognition Algorithm

In general, our recognition algorithm is scanline based. In order to improve robustness, we decided to not only use a single scanline, but a set of multiple, potentially arbitrarily oriented scanlines (see figure 1). If multiple scanlines cross the bar code, each with a different sensitivity, we can increase the chances that at least one of them will result in a properly recognized code. Also, multiple scanlines can be combined in a majority-voting fashion, were inaccuracies due to dirt or reflections on one line can be compensated by two or more correct identifications on other lines. By applying slightly different recognition parameters along each individual scanline (i.e., the binarization threshold that categorizes pixels into either black or white), the overall recognition accuracy can also be improved. Last not least, by using a variable amount of scanlines, we have a simple mechanism to adapt our algorithm to the processing power of the individual phone it is running on: The more computational capabilities available, the more scanlines and orientations[10] we can

[9] See `sun.com/software/communitysource/j2me/mmapi/`.

[10] Initial tests show, however, that most users actually take care to properly align the bar code when using their camera phones, though right-angled rotations (i.e., 90, 180, and 270 degrees) were more common. Especially superimposing our scanlines on the mobile phone's viewfinder would further guide users to a proper alignment.

Figure 3: *Successfully recognized codes:* The above pictures show that our algorithm can handle reflections and shadows as well as slightly crumpled paper.

Figure 4: *Unrecognized codes:* The above three codes are examples of failed recognition attempts. The leftmost image shows a too crumpled bar code; the middle one is angled just too much (we only used horizontal scanline in this experiment); the rightmost image is too blurry.

try. Since the algorithm is scanline based, it cannot cope as well with image distortions as transformation-based algorithms. However, as the analysis below will show, our implementation is sufficiently robust even for lower image resolutions. Also, it is quite fast, has very little memory requirements, and can be implemented relatively easy.

3.2 Performance Evaluation

We have analyzed the recognition performance of our algorithm along two axes: *focus* and image *resolution*, as these are currently the two most important parameters influencing recognition accuracy on a mobile camera phone. The camera focus directly affects a picture's sharpness. Results indicate that focus remains a problem, while low camera resolutions such as 640x480 pixels are not critical.

In order to allow camera phones to scan a bar code from close-up, two options are available. Increasingly, camera phones are being equipped with auto-focus lenses that have been developed over the last several years [Tur05, Chu05]. As of spring 2006, most major handset manufacturer offer auto-focus models (e.g., the Nokia N90, SonyEricsson's K790 and W810, or Samsungs SCH series). Models that still use fixed focus lenses need to be adopted to the required short distance with the help of a macro lens, a cheap accessory that is carried by many mobile phone dealers. Ideally, our system would be deployed on auto-focus systems, thus eliminating the need for any specific hardware accessory. However, macro lenses are cheap and unobtrusive enough to make their use in a large-scale trial seem feasible.

For our evaluation, we took pictures of ten bar codes situated on common retail goods. From each code we took 16 pictures, starting with a distance of 10 cm between the code and the camera, and decreasing it in steps of 0.5 cm down to a minimum distance of 2 cm.

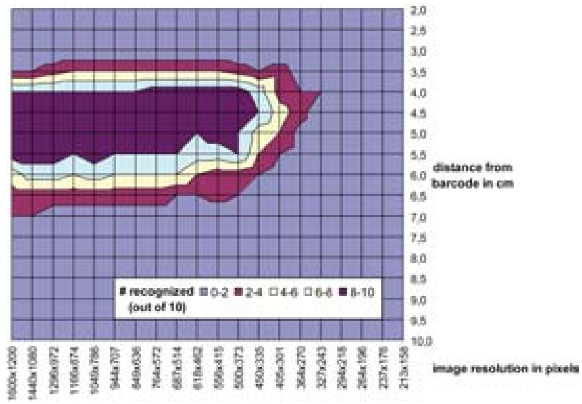

Figure 5: *Focus and resolution influence:* The above graph shows the combined influence of both *focus* and *resolution* on the recognition rate when using a macro lens. The innermost dark area indicates an acceptable recognition rate and is situated at about 4–4.5 cm distance between camera and bar code, at a resolution of at least 640x480 pixels.

We used a Nokia N90 mobile phone to take all the pictures with a maximal resolution of 1600x1200 pixels. Since the auto-focus feature of this phone currently can't be activated neither from a J2ME Midlet nor from a C++ Symbian application, due to the missing functionality in the free libraries provided by Nokia, we used an attached macro lens. From each of the 16 images that were taken at different distances, we created 20 versions in different resolutions (using a desktop image manipulation program). This resulted in a set of 320 images for each bar code. Figure 5 visualizes the influence of the image resolution and camera focus on the recognition success. Results indicate that focus remains a problem, while today's camera phone resolutions of about 1 MPixel are clearly sufficient for reliable bar code recognition. As pointed out above, however, we expect the market soon to adopt auto-focus not only for high-end devices, but also for common models due to recent technological advances [Chu05, MDK$^+$06].

4 Prototypical Applications

In order to illustrate the use of potential of out toolkit, we implemented and provide two prototypical applications. The first prototype represents a simple literature information system, providing information about scanned books, such as their current price or a list of related items (see figure 6).

The second prototypical application implements a tool for checking ingredients in nutrition products that could trigger an allergic reaction. Using a retail goods database such as GS1[11] we could gain access to detailed allergen information of individual products, based on their EAN-13 code. Together with an individual shopper's allergy profile, the

[11] See www.gs1.org

Figure 6: *Example application:* The above screenshots show a literature information system – sample application implemented using our recognition and resolution system.

application is able to warn the user of potential allergic reactions to an item with a single click.

Given our toolkit, the implementation effort for these two demos was quite low. Changes on the provided J2ME client were limited to renaming issues and took about 5 minutes. Implementing the allergy test application required the implementation of a "service connector" (cf figure 2), providing access to a (commercial) product database, and the implementation of the service component that would generate the result according to a previously defined user profile.

5 Conclusion

Linking the physical world with virtual information is a powerful concept. Even though RFID promises to provide an easy-to-use, pervasive linkage that could easily be accessed with the help of small mobile devices (such as NFC-enabled mobile phones), their use on everyday items such as soda cans or cereal boxes remains unlikely for the next several years. In contrast, regular 1D-bar codes (EAN-13) are ubiquitous – printed on billions of products worldwide and already linked to a wealth of both free and commercial databases. So far, consumer camera phones have in general only be able to recognize specific 2D bar codes, which – just as RFID tags today – have not yet been widely adopted. With the increasing availability of high resolution camera phones, as well as the prospect of cheap auto-focus lenses, using mobile phone to tap into the wealth of EAN-13 product information becomes feasible. This opens up novel ways of fielding systems that explore the "internet of things" – not just for toy applications or small, specially equipped user bases, but for any user with a conventional camera phone, for almost any product, nearly everywhere. In this paper we present the necessary tools for the easy and fast creation and prototyping of own services and applications based on EAN-13 recognition and resolution. They are freely available for download from `http://batoo.sourceforge.net/`. We hope that this "low tech" version of *bridging the gap* will allow the community to quickly develop and try out more realistic and widespread applications, and thus gain real-world experiences for better jump-starting the future "internet of things", today.

References

[BM98] Rob Barrett and Paul P. Maglio. Informative things: how to attach information to the real world. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 81–88, New York, NY, USA, 1998. ACM Press.

[CH05] Douglas Chai and Florian Hock. Locating and Decoding EAN-13 Barcodes from Images Captured by Digital Cameras. Addendum to Proceedings ICIS2005, 2005.

[Chu05] Myung-Jin Chung. Development of compact auto focus actuator for camera phone by applying new electromagnetic configuration. In Kee S. Moon, editor, *Proceedings of SPIE. Volume 6048 – Optomechatronic Actuators and Manipulation*, pages 152–160 (60480J), Sapporo, Japan, December 5, 2005. Intl. Society for Optical Engineering.

[Jue05] Ari Juels. RFID Privacy: A Tecnical Primer for the Non-Technical Reader. In Katherine Strandburg and Daniela Stan Raicu, editors, *Privacy and Technologies of Identity: A Cross-Disciplinary Conversation*. Springer, 2005.

[KBM+02] Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Debbie Caswell, Philippe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris, John Schettino, Bill Serra, and Mirjana Spasojevic. People, places, things: web presence for the real world. *Mobile Networks and Applications*, 7(5):365–376, 2002.

[MDK+06] Peter M. Moran, Saman Dharmatilleke, Aik Hau Khaw, Kok Wei Tan, Mei Lin Chan, and Isabel Rodriguez. Fluidic lenses with variable focal length. *Applied Physics Letters*, 88(041120), January 23, 2006.

[MJO99] Ruben Muniz, Luis Junco, and Adolfo Otero. A robust software barcode reader using the Hough transform. In *Proceedings of the 1999 International Conference on Information Intelligence and Systems*, pages 313–319, 1999.

[OHH04] Eisaku Ohbuchi, Hiroshi Hanaizumi, and Lim Ah Hock. Barcode Readers using the Camera Device in Mobile Phones. In *CW*, pages 260–265. IEEE Computer Society, 2004.

[PKA05] Lauri Pohjanheimo, Heikki Keränen, and Heikki Ailisto. Implementing Touchme Paradigm with a Mobile Phone. In *sOc-EUSAI '05: Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence*, pages 87–92, New York, NY, USA, 2005. ACM Press.

[RA00] Jun Rekimoto and Yuji Ayatsuka. CyberCode: designing augmented reality environments with visual tags. In *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*, pages 1–10, New York, NY, USA, 2000. ACM Press.

[Roh04] Michael Rohs. Real-World Interaction with Camera-Phones. In *2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*, pages 39–48, Tokyo, Japan, November 2004.

[Tur05] Emily Turrettini. Korean Start-Up Develops Auto-Focus in Camera Phone. Weblog. Available at `www.textually.org/picturephoning/archives/2005/02/007111.htm`, February 2005.

[WFGH99] Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. Bridging physical and virtual worlds with electronic tags. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 370–377, New York, NY, USA, 1999. ACM Press.